ARMY RESEARCH LABORATORY

Contract Req
Software Requirements Analysis
Version 1: A C-BASS Component

by Denis McGurin

ARL-MR-458   August 1999

19990819 088

DTIC QUALITY INSPECTED 4

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

Contract Req Software Requirements Analysis Version 1: A C-BASS Component

Denis McGurin
Corporate Information and Computing Directorate, ARL

Abstract

This document contains the software requirements analysis for a prototype of Contract Request Version 1.0 (Contract Req). As a component of the Corporate Business Application Software System (C-BASS), this application automates the preparation of the requester's Procurement Data Package (PDP). The document follows the process of structured analysis, or step-wise refinement of requirements, as applied to the development of Contract Req. The "environmental model" includes a high-level system description, followed by a context diagram and a list of events to which the system must respond. The "behavioral model" includes a data flow diagram (DFD) for each of the five Contract Req subsystems. From this representation, the basic functional specifications are derived and represented in structured English (or program design language). The final segment of the document includes a data dictionary.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

Contract Request Vision 1.0 is a component of the Corporate Business Application Software System (C-BASS) cluster of applications, an integrated set of Lotus Notes and Web-based software to support U.S. Army Research Laboratory (ARL) electronic workflow and task automation. (Throughout this report, Contract Request Version 1.0 is abbreviated as Contract Req.) The motivating force behind this work effort has been ARL downsizing and the findings contained in the Business Process Reengineering (BPR) report on the contracting process published in 1995 [1]. The BPR "To-Be Model: Formal Contracts" [1] identified potential process improvements — some of which require computer automation — that would increase productivity of contracting operations at ARL. Development of a full-production Contract Req will proceed in phases, using an incremental, evolutionary approach. The system described in this document is the first version in this process.

1.1 Contract Req Prototype. The purpose of the Contract Req prototype is to provide a secure client/server system that provides for the processing of contract requests. This proof-of-principle prototype will alleviate some of the risks involved in implementing new technologies used to build the ARL Intranet. The project will also refine requirements described in the ARL BPR To-Be Model [1].

1.2 Development Plan and Project Schedule. The "Contract Req Software Development Plan" [2] states a definition of the problem; gives an overview of technical, management, and reliability issues; and provides a detailed project schedule. The report given herein covers the work accomplished to solidify user requirements (primarily drawn from existing high-level design documents) and the analytical expansions used to derive a data flow model, a pseudocode representation of processing, and a data dictionary.

1.3 Contents of the Report. This document presents the results of a structured analysis used to derive the software requirements for Contract Req, starting with the baseline given in the BPR "To-Be Model: Formal Contracts" [1]. The body of the report contains five sections:

- "Structured Analysis Overview" - briefly explains the methodology used to extract the functional specifications.

- "System Overview" - delineates the basic Contract Req concept and outlines the high-level requirements.

- "System Requirements" - breaks the more generic statements into low-level, derived requirements and describes each in detail.

- "Functional Specifications" - discusses the products of the structured analysis (i.e., the data flow diagrams and the structured English narrative) for each subsystem of Contract Req.

- "Data Dictionary" - lists each of the Contract Req data elements, giving a full description and type for the data model.

2. Structured Analysis Overview

Modern software engineering utilizes structured analysis [3] as a powerful methodology for developing system specifications. Through a series of step-wise refinements, detailed delineation of the system's components and their behavior are extracted from high-level descriptions of system features and functions. In other words, primary system elements are broken down into progressively more detailed levels of processes, and the data paths between these processes are defined. Three modeling tools facilitate this decomposition: (1) data flow diagrams (DFDs), (2) structured English process narratives (either pseudocode or program design language [PDL]), and (3) a data dictionary. Accuracy and precision in progressively expanding design definitions are critical to successful system development.

The results of this analytical approach are systematic elaborations of product requirements, typically expressed as parts of two separate models:

2

- An environmental model that defines the system's interfaces to the outside world (see section 3, "System Overview").

- A behavioral model that defines the internal behavior the system must exhibit in order to deal with the environment (see section 4, "System Requirements," and section 5, "Functional Specifications").

3. System Overview

The environmental model typically consists of three components: (1) a concise statement of the system's purpose or required functionality, (2) a context diagram, and (3) an event list. The context diagram is the highest level DFD. It shows the system as a single process, including user interaction and communication with external systems, as well as data flow input and output. The event list provides an index of outside stimuli to which the system responds.

3.1 Purpose and Required Functionality. As a system, the Contract Req prototype provides a secure, automated means for the preparation, submission, approval, and tracking of certain aspects of the contracting process. The complete contract life cycle is complex and multistaged. Contract Req currently automates only the preparation of the requester's Procurement Data Package (PDP). See "Contract Req Software Development Plan" [2] for an explanation of the exact portions of the contract process automated by this product.

Table 1 lists the high-level requirements for the Contract Req prototype and gives a general description of what the requirement involves.

Figure 1 shows the context diagram for the prototype. This diagram displays the external entities (e.g., users, functional areas, and legacy systems), represented by the squares, with which Contract Req communicates. The data flowing into and out of Contract Req are represented by the arrows. A few elements on the DFD need additional explanation. First, the external entity "User" represents all users of the system. The data flow associated with this box is limited to

3

Table 1. High-Level Requirements for the Contract Req Prototype

Requirement	General Description
Security	Provide security measures to prevent unauthorized access to the system and its data, and keep authorized users from performing tasks not allowed in their roles.
Contract request preparation	Provide a means for the requesters and functional users to input/edit relevant information pertaining to a contract request.
Automated request routing	Automate the process of routing contract requests to the various functional areas.
Electronic approval	Provide a means for approving officials and functional users to electronically approve or reject a contract request.
Request tracking	Allow users to track the status of active contract requests currently in the system.
Legacy system interface	Implement automated interfaces to Standard Operations and Maintenance Army Research Development System (SOMARDS).
Reporting	Provide users and management with a means for reporting cycle times and costs.
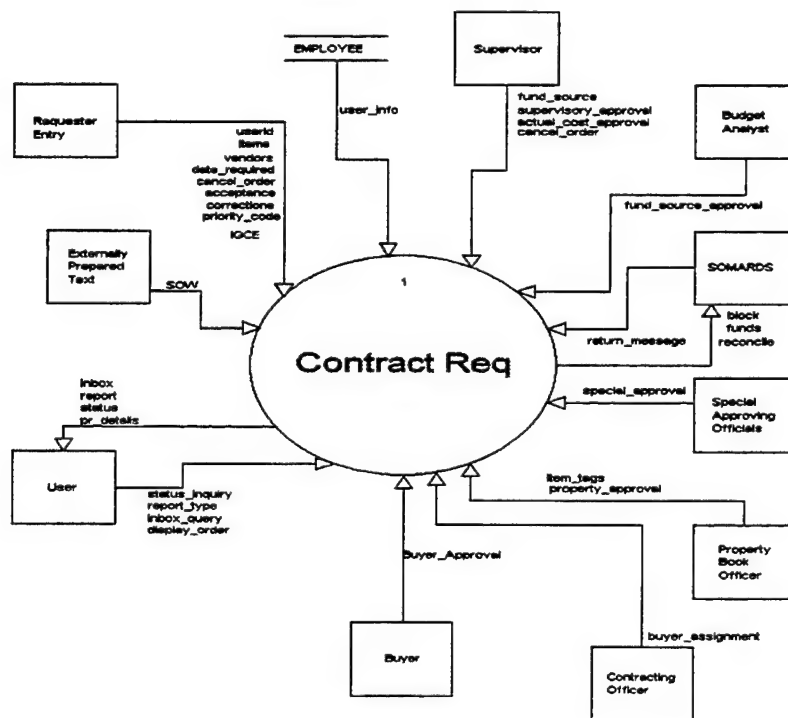
Figure 1. Context Diagram for Contract Req.

4

display of information. All the other external entities (e.g., "Requester," "Supervisor," etc.), and their corresponding data flows, show the specific information that is passed to Contract Req, either by that user or by the system. The data store EMPLOYEE contains user information such as name, phone number, address, office symbol, etc.

3.2 Event List. The following are the events to which the system must respond.

- Requester initiates new contract request.
- Requester prepares request.
- Requester corrects rejected request.
- Requester, supervisor, or budget analyst completes fund source.
- Requester or supervisor cancels request.
- Budget analyst certifies fund source.
- SOMARDS certifies and commits funds.
- Special approving official(s) approves request item(s).
- Property book officer approves request.
- Contracting officer assigns buyer to request.
- Supervisor approves actual costs.
- User submit status inquiry.
- User request report.

4. System Requirements

Antecedent studies and legacy systems also contribute to Contract Req's requirements. The "To-Be Model: Formal Contracts" [1] was produced during the BPR study in 1995. However, for some areas, this document lacks detail, making it necessary to derive missing elements. Additionally, the limited scope of a prototype necessitates leaving out some of the more complicated or vague automation requirements of the "To-Be Model" or where the requirement already exists in a Department of Defense (DOD) standard system (e.g., functionalities handled by the Standard Army Automated Contract System [SAACONS]). The "Contract Req Software

Development Plan" [2] more fully addresses the constraints on the prototype and the requirements of the legacy systems.

The following elaboration more fully defines the proscribed system tasks enumerated in Table 1.

E1

E2   Security

    E21   Prevent unauthorized access

    Description — Prevent unauthorized access to the system and its data.

    Source — Derived, due to the nature of the system.

    Interfaces to major functions and external entities:

        User

    E22   Enforce role restrictions

    Description — Prevent users from performing tasks or accessing/editing data that are out of the scope of their role.

    Source — BPR "To-Be Model" document, requirement A151.

    Interfaces to major functions and external entities:

        User

        Approvals

        Edits

        Employee address book (for roles).

E3   Contract request preparation

    E31   Create new contract request

    Description — Allow the requester to create a new contract request with preliminary user information filled in.

    Source — BPR "To-Be Model" document, Automation Requirements section, requirement A151

    Interfaces to major functions and external entities:

User

Security

Employee address book (for user info)

E32   Select items

Description — Provide a means for the requester to enter item descriptions, specifications, quantities, and estimated costs.

Source — BPR "To-Be Model" document, "Automation Requirements" section, requirement A151.

Interfaces to major functions and external entities:

User

Security

E33   Complete contract request

Description — Provide a means for the requester and/or approving supervisor to complete the contract request.

Source — BPR "To-Be Model" document, Automation Requirements section, requirements A151.

Interfaces to major functions and external entities:

User

Security

E34   Item tag input

Description — Provide a means for the Property Book Officer to enter item tags.

Source — BPR "To-Be Model" document, Automation Requirements section, requirements A151.

Interfaces to major functions and external entities:

User

Security

E35   Edit contract request

Description — Provide a means for users to edit certain request details as needed.

Source — Derived, due to the need for making corrections to rejected contract request.

Interfaces to major functions and external entities:

7

User

Security

E4

E5   Automated routing

Description — Automate the process of routing contract requests to the various functional areas and approving officials.

Source — BPR "To-Be Model" document, Automation Requirements section, requirements A151.

Interfaces to major functions and external entities:

Security

Employee address book (for default routing)

E6   Electronic approval

Description — Provide a means for approving officials and functional users to approve or reject a contract request.

Source — BPR "To-Be Model" document, Automation Requirements section, requirements A151.

Interfaces to major functions and external entities:

User

Security

E7   Request tracking

Description — Allow users to track the status of active contract requests currently in the system.

Source — User requested.

Interfaces to major functions and external entities:

User

E8 **Legacy system interfaces**

 E81 **Budget legacy system interface**

  Description — Provide an electronic interface to the legacy financial system (SOMARDS) that automates the certification and commitment of funds.

  Source — "To-Be Model" section, process model diagram A12, process A153.

E9

E10 **Reporting**

 Description — Provide users and management with a means for reporting cycle times and costs.

 Source – Derived from ARL corporate experience. Not specifically noted in BPR Formal Contracts documentation.

 Interfaces to major functions and external entities:

  User

  Security

E11 **Navigation**

 Description — Provide users with a means for navigating to the various functional areas within the system.

 Source — Derived from the requirements listed previously.

 Interfaces to major functions and external entities:

  User

5. Functional Specifications

The behavioral model expands the results from the environmental model to more fully define how the system performs prescribed tasks. Typical representations in this mode are (1) concise flow charts showing how information is transformed as it moves through the system and subsystems; (2) a set of structured English statements forming a processing narration based on

data types, control structures, and transformations; and (3) a data dictionary defining each data item.

5.1 Contract Req Subsystems. The eight functions listed in the previous section identify the major components of Contract Req: (1) security, (2) contract request preparation, (3) automated routing, (4) electronic approval, (5) request tracking, (6) legacy system interfaces, (7) reporting, and (8) navigation. These categories consolidate into five subsystems: (1) prepare contract request, (2) approve funds, (3) obtain approvals, (4) edit contract requests, and (5) inquire about status.

Figure 2 is a DFD showing the major functional subsystems of Contract Req. Each of the five bubbles in the diagram represents a major subsystem or process of Contract Req, with the arrows showing the data flowing into and out of the processes.
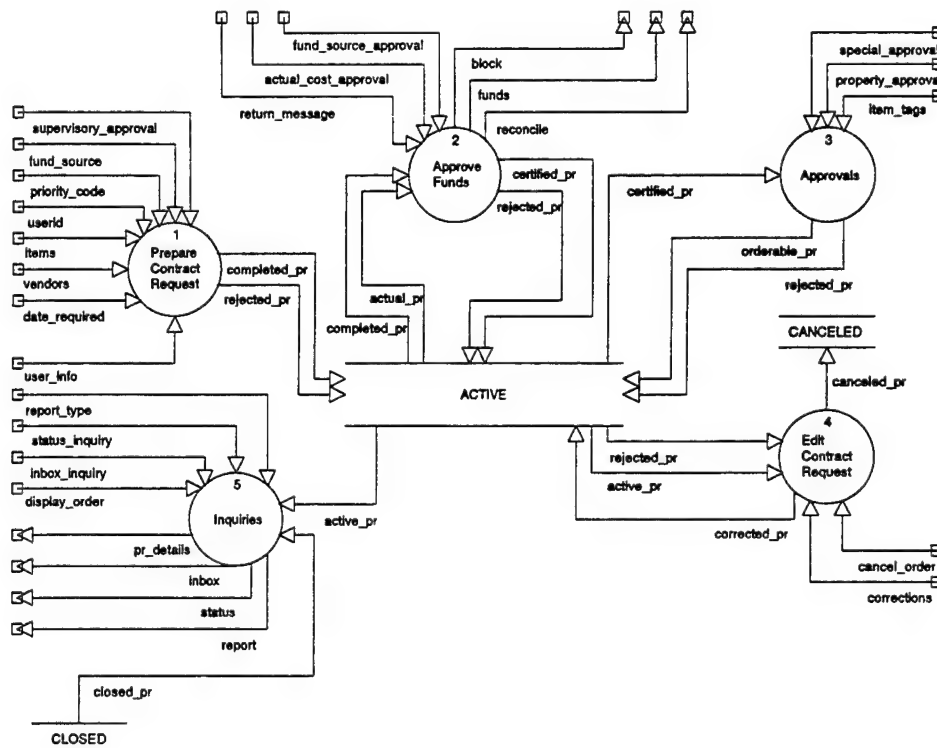

Figure 2. Major Subsystems of Contract Req.

The data store ACTIVE—a Lotus Notes database located in the center of the diagram—holds all the active contract requests, each waiting for the various users to perform their functions on them. The CLOSED and CANCELED data stores contain contract requests that have been closed out and canceled, respectively. The small squares along the outer edges of this DFD are interfaces to the outside world.

No process bubble for security appears at this level because the Lotus Notes application environment handles security and verifies user authorization. Additionally, enforcement of role restrictions are handled within each subsystems, as detailed in the following sections.

5.2 Subsystems Data Flow Diagrams. System objects and operations can be coherently represented as DFDs. A DFD can be used to capture system concepts and components at any level of abstraction. Each of the following five DFDs (Figures 3–7) provides more detail for the information flow and functionality of each of the identified Contract Req subsystems.

Figure 3 shows the DFD for the "Prepare Contract Request" subsystem. The major inputs to this process (and its basic functions) are the requester information (derived from the user supplied userid and the EMPLOYEE data store), item details, vendor information (if known at this time), and the date by which the items are required. The fund source completes the information for the request, and the supervisory approval puts the request into the contracting cycle.

Figure 4 shows the DFD for the "Approve Funds" subsystem. This subsystem, besides having interfaces to users for approvals, also has connections to the SOMARDS legacy system. The "Build Block" process is executed at the start of the day and creates the transaction block that will be used by Contract Req for the remainder of the day. As eligible contract requests are created during the course of the day, the "Certify Funds" process queries SOMARDS and grabs the returning message. Depending on the results of this query, the request is either certified or rejected (with explanation). At the end of the day, the "Reconcile" process is executed to balance the transaction block.

11
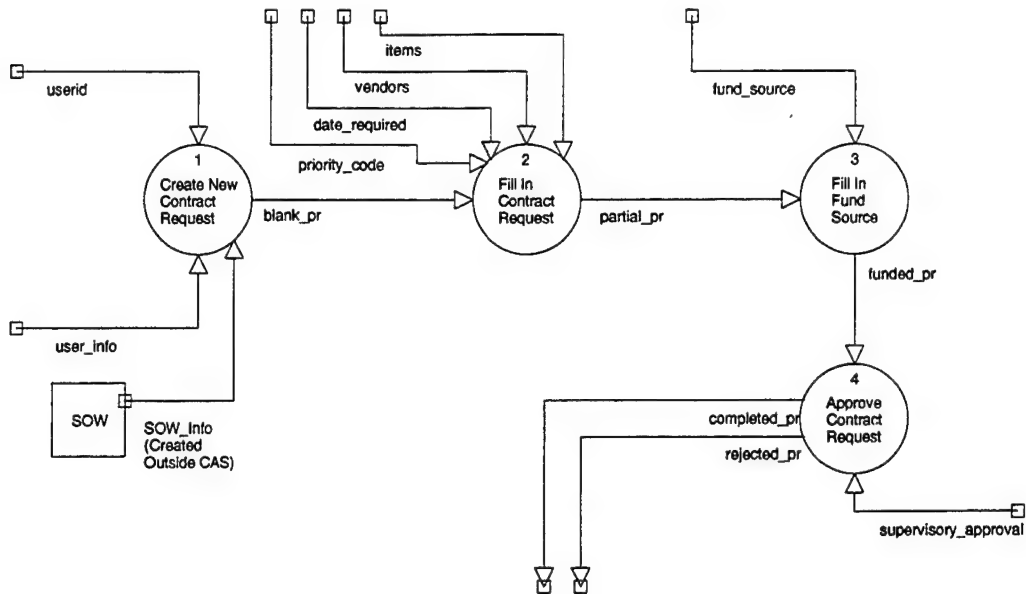
Figure 3. Prepare Contract Request Process.
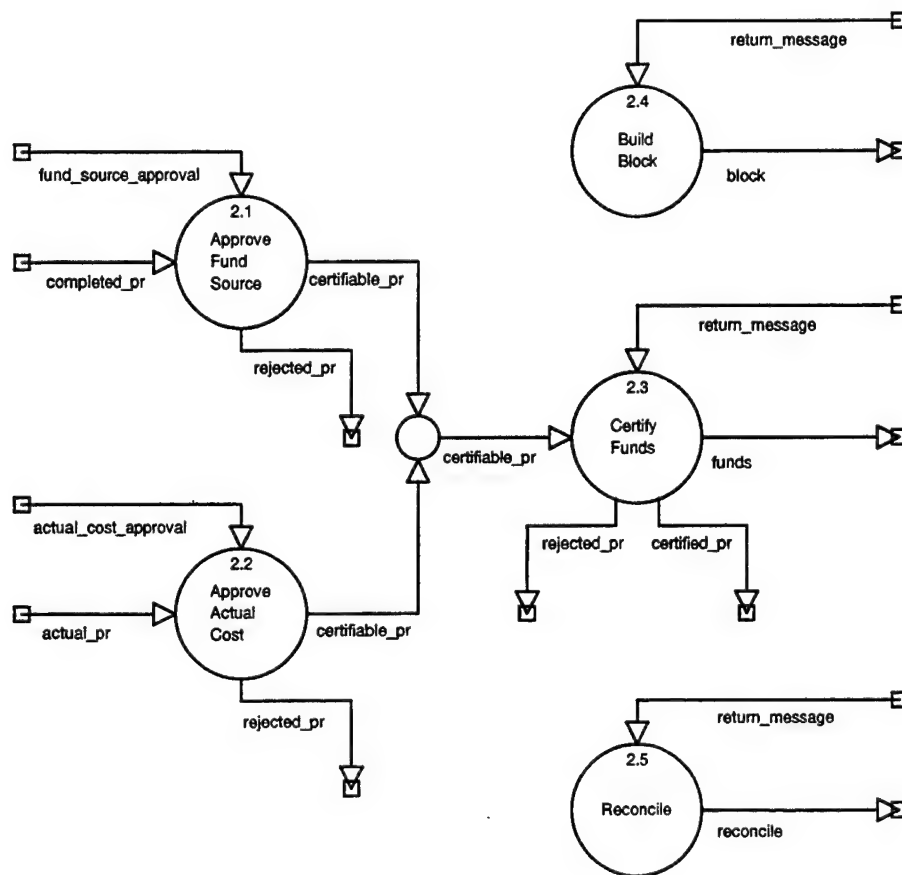
Figure 4. Approve Funds Process.

12

The "Approvals" process is shown in Figure 5. At this point, the various approving officials attach their approval or rejection to the request. The property book officer also attaches item tags to the individual items in order to flag them during receipt of the shipment.
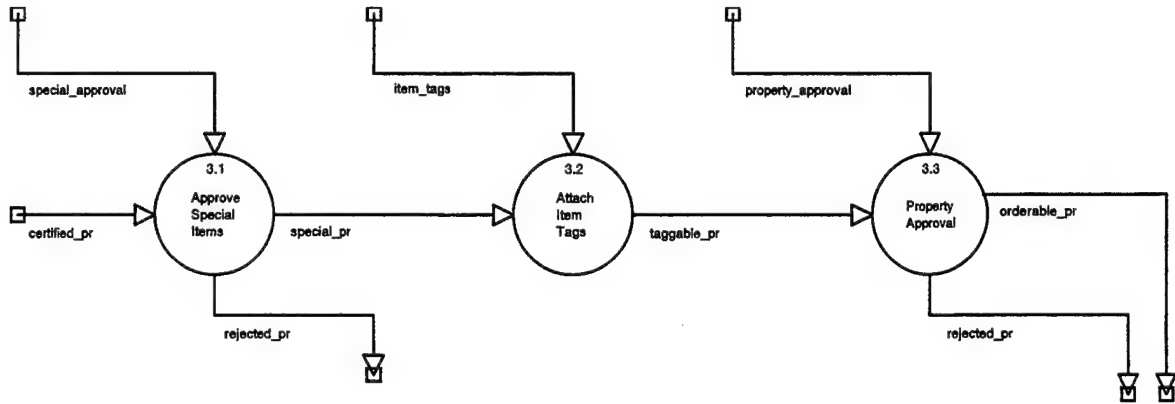
Figure 5. Approvals Process.

Figure 6 diagrams the "Edit Contract Request" process. The rejected contract request is displayed for the requester, who then enters the corrections. What fields within the request the user can edit depends on where the rejection came from and how far along in the approval process the request has traveled.
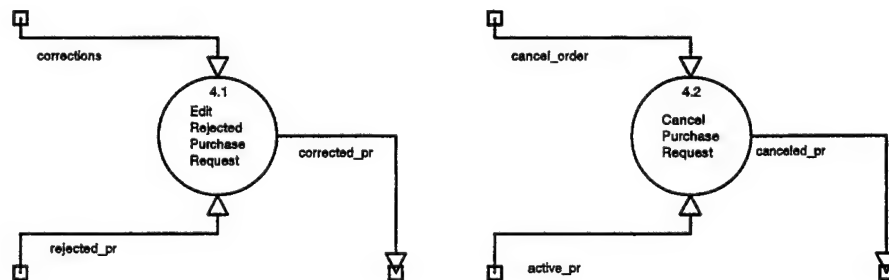
Figure 6. Edit Contract Request Process.

The "Inquiries" process is diagramed in Figure 7. The processes shown in this figure are used to display to the user (1) pending actions (inbox), (2) contract request details, (3) the status of requests, and reports.

13
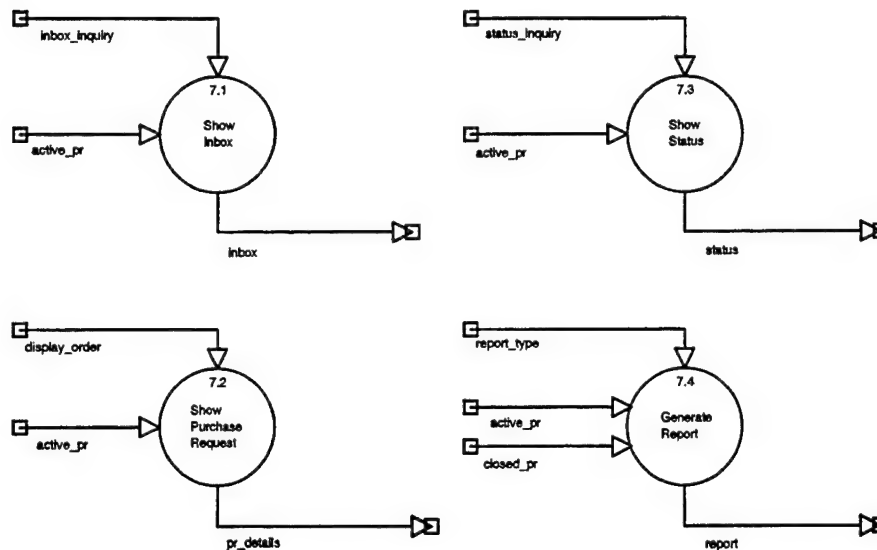
Figure 7. Inquiries Process.

5.3 Processing Narration. Having captured the flow of information and identified data objects, each transformation can be further expanded by using the stylized notation of structured English. Basic procedural constructs are combined with English phrases to give a concise description for each major operation listed in the prescribed tasks analysis given in section 4.

E1   Security

   E11   Login

> *Input* — userid, passwd
>
> *Process:*
>
>> REPEAT
>>
>>> GET from user the **userid, passwd**
>>
>> UNTIL VALID **userid, passwd**
>>
>> ALLOW login
>
> *Output* – Access on Success, Error Message on Failure

   E12   Role

> *Input* — role, action
>
> *Process*
>
>> GET from **EMPLOYEE** the **role** using **requester_userid**

14

IF VALID **action** for **role** THEN

    EXECUTE **action**

ELSE

    NULL

ENDIF

Output — action_results

E2   Prepare contract request for ordering

E21   Create new contract request

Input — pr, requester_userid, user_info

Process 1.1

    GET from **EMPLOYEE** the **user_info** using

                requester_userid

    SET in **pr** the **requester_userid**

    SET in **pr** the **user_info** using **user_info**

Output — blank_pr

E22   Fill in contract request

Input — blank_pr, priority_code, items, vendors, date_required

Process 1.2

    GET from user the **priority_code**

    SET in **pr** the **priority_code**

    DO WHILE there is another **item** to add

        GET from user the **item**

        SET in **pr** the **item**

    ENDDO

    GET from user the **specifications**

    SET in **pr** the **specifications**

    DO WHILE there is another **vendor** to add

        GET from user the **vendor**

        SET in **pr** the **vendor**

    ENDDO

GET from user the **date_required**

SET in **pr** the **date_required**

Output — partial_pr

E23 Fill in fund source

Input — partial_pr, fund_source

Process 1.3

GET from user the **fund_source**

SET in **pr** the **fund_source**

Output — funded_pr

E24 Correct contract request

Input — rejected_pr, corrections

Process 4.1

DISPLAY to user **rejected_pr** and **explanation**

DO WHILE there are more **corrections**

GET from user the **corrections**

SET in **pr** the **corrections**

ENDDO

Output — corrected_pr

E25 Cancel request

Input — active_pr, cancel_order

Process 4.2

DISPLAY to user the **active_pr**

GET from user the **cancel_order**

PUT **canceled_pr** into **CANCELED**

Output — cancelled_pr

E3 Routing

E31 Automated routing

Input — active_pr

Process — TBD

Output — active_pr

16

E32   Manual routing

>   *Input* — active_pr

>   *Process* — **TBD**

>   *Output* — active_pr

E33   Assign Buyer

>   *Input* — orderable_pr

>   *Process 5.1*

>>   DISPLAY to user the **orderable_pr**

>>   GET from user the **buyer_assignment**

>>   SET in **pr** the **buyer_assignment**

>>   SET in **pr** the **inbox_location** to buyer

>   *Output* — assigned_pr

E4   Approvals

E41   Supervisory approval

>   *Input* — funded_pr, supervisory_approval

>   *Process 1.4*

>>   DISPLAY to user the **funded_pr**

>>   GET from user the **supervisory_approval**

>>   IF **supervisory_approval** is 'Yes' THEN

>>>   SET in **pr** the **supervisory_approval** to 'Yes'

>>>   SET in **pr** the **request_date** to today's date

>>>   SET in **pr** the **inbox_location** to budget

>>   ELSE

>>>   GET from user the **explanation**

>>>   SET in **pr** the **supervisory_approval** to 'No'

>>>   SET in **pr** the **explanation**

>>>   SET in **pr** the **inbox_location** to requester

>>   ENDIF

>   *Output* — completed_pr, rejected_pr

17

E42　Fund source approval

Input — completed_pr, fund_source_approval

Process 2.1

    DISPLAY to user the **completed_pr**

    GET from user the **fund_source_approval**

    IF **fund_source_approval** is 'Yes' THEN

        SET in **pr** the **fund_source_approval** to 'Yes'

        SET in **pr** the **inbox_location** to certification

    ELSE

        GET from user the **explanation**

        SET in **pr** the **fund_source_approval** to 'No'

        SET in **pr** the **explanation**

        SET in **pr** the **inbox_location** to requester

    ENDIF

Output — certifiable_pr, rejected_pr

E43　Special approval

Input — certified_pr, special_approval

Process 3.1

    DISPLAY to user the **certified_pr**

    GET from user the **special_approval**

    IF **special_approval** is 'Yes' THEN

        SET in **pr** the **special_approval** to 'Yes'

        SET in **pr** the **inbox_location** to

                property_book_officer

    ELSE

        GET from user the **explanation**

        SET in **pr** the **special_approval** to 'No'

        SET in **pr** the **explanation**

        SET in **pr** the **inbox_location** to requester

    ENDIF

Output — special_pr, rejected_pr

E44   Property approval

Input — special_pr, property_approval

Process 3.3

DISPLAY to user **special_pr**

GET from user **property_approval**

IF **property_approval** is 'Yes' THEN

SET in **pr** the **property_approval** to 'Yes'

SET **inbox_location** to contracting_officer

ELSE

GET from user **explanation**

SET in **pr** the **property_approval** to 'No'

SET in **pr** the **explanation**

SET in **pr** the **inbox_location** to requester

ENDIF

Output — orderable_pr, rejected_pr

E45   Actual cost approval

Input — actual_pr, actual_cost_approval, explanation

Process 2.2

DISPLAY to user **actual_pr**

GET from user **actual_cost_approval**

IF **actual_cost_approval** is 'Yes' THEN

SET in **pr** the **actual_cost_approval** to 'Yes'

SET in **pr** the **inbox_location** to buyer

ELSE

GET from user **explanation**

SET in **pr** the **actual_cost_approval** to 'No'

SET in **pr** the **explanation**

SET in **pr** the **inbox_location** to requester

ENDIF

Output — certifiable_pr, rejected_pr

19

E46   Buyer approval

>*Input* — assigned_pr, buyer_approval, explanation
>
>*Process 2.2*
>
>>DISPLAY to user **assigned_pr**
>>
>>GET from user the **buyer_approval**
>>
>>IF **buyer_approval** is 'Yes' THEN
>>
>>>SET in **pr** the **buyer_approval** to 'Yes'
>>
>>ELSE
>>
>>>GET from user **explanation**
>>>
>>>SET in **pr** the **buyer_approval** to 'No'
>>>
>>>SET in **pr** the **explanation**
>>>
>>>SET in **pr** the **inbox_location** to requester
>>
>>ENDIF
>
>*Output* — approved_pr, rejected_pr

E5   Interface with legacy systems

E51   Build block

>*Input* — N/A
>
>*Process 2.3*
>
>>PUT to **SOMARDS** the **block**
>>
>>GET from **SOMARDS** the **return_message**
>
>*Output* — N/A

E52   Certify funds

>*Input* — certifiable_pr
>
>*Process 2.4*
>
>>GET from **certifiable_pr** the **funds**
>>
>>PUT to **SOMARDS** the **block, funds**
>>
>>GET from **SOMARDS** the **return_message**
>>
>>IF **return_message** is 'OK' THEN
>>
>>>SET in **pr** the **certification** to 'Yes'

20

SET in **pr** the **inbox_location** to special_approval

ELSE

SET in **pr** the to **certification** to 'No'

SET in **pr** the **explanation** to **return_message**

SET in **pr** the **inbox_location** to requester

ENDIF

Output — certified_pr, rejected_pr

E53   Reconcile

Input — N/A

Process 2.5

PUT to **SOMARDS** the **block, reconcile**

GET from **SOMARDS** the **return_message**

Output — N/A

E54   Upload to SAACONS

Input — assigned_pr

Process 5.1

GET from **assigned_pr** the **upload**

PUT to **SAACONS** the **upload**

Output — N/A

E6   Status inquires

Input — active_pr

Process 7.3

GET current **status** from **ACTIVE**

DISPLAY to user the **status**

Output — status

E7   Generate reports

Input — report_type

Process 7.4 — TBD

Output — report

21

E8   Navigation

    E81   Navigate

        *Input — **TBD***

        *Process — **TBD***

        *Output — **TBD***

    E82   Logout

        Input — N/A

        *Process — **TBD***

        Output — N/A

6. Data Dictionary

While DFDs and pseudocode (structured English) are important to system specifications, additional information is required for a complete analytical model. The content of each data or control item should be more fully identified. A data dictionary is a quasi-formalism for describing content of information as it flows through the system. The standard notation conventions are

Notation	**Meaning**
=	is composed of
+	and
[|]	either - or
{ }^{n}	n repetitions of
( )	optional data
* *	comments

Contract Req prototype's data dictionary appears in following text. Each left-hand element is taken from the DFD and the PDL representations of the system. These data items are then given an expanded, unambiguous definition in the right-hand column.

22

acceptance =	*requester accepts or returns shipment item* ["Yes" | "No"]
ACTIVE =	{active_pr}
action =	** *TBD*
action_results =	** *TBD*
active_pr =	*purchase request at some point in the approval cycle*
actual_cost =	*actual cost of an item* *units: dollars*
actual_cost_approval =	*supervisory approval of the total actual cost of the request* ["Yes" | "No"]
actual_pr =	*purchase request where the total actual cost is greater than the total estimated cost* approved_pr + {actual_cost} + {item_tot_act_cost} + total_actual_cost
address =	** street_address + (mail_stop) + city + state + postal_code + (country)
approved_pr =	*purchase request that has procurement buyer approval* assigned_pr + buyer_approval
assigned_pr =	*purchase request that has been assigned to a buyer by the contracting officer* orderable_pr + buyer_assignment
bar_code_no =	*property book bar code number* {alphanumeric}
batch_no =	*SOMARDS batch number* "Contract Req"
blank_pr =	*purchase request with the requester and delivery info filled* requester_userid + user_info

bldg_no =	*building number* {alphanumeric}
blk_no =	*SOMARDS block number* "ARL "
blk_tkt_dt =	*SOMARDS block ticket date* *format: MMDDYY* date
block =	*SOMARDS build block data* trns_cd + user_auth_key + cmd_dsg + update_code + blk_no + blk_tkt_dt + tot_blk + batch_no + tot_batch
buyer_approval =	*buyer approval of purchase request* ["Yes"	"No"]
buyer_assignment =	** buyer_userid + saacons_buyer_code
buyer_userid =	** userid
cancel_order =	*order to cancel purchase request* ["Yes"	"No"]
CANCELED =	{canceled_pr}
canceled_pr =	*purchase request that has been canceled* active_pr + cancel_order
certifiable_pr =	*purchase request that has fund source approval or actual costs approved* [completed_pr + fund_source_approval	actual_pr + actual_cost_approval]
certification =	*SOMARDS certification* ["Yes"	"No"]
certified_pr =	*purchase request that has been certified by SOMARDS* certifiable_pr + certification
city =	** {alphabetic_character}

24

CLOSED =	{closed_pr}
closed_pr =	*purchase request that has been accepted by the requester* tagged_pr + {acceptance}
cmd_dsg =	*SOMARDS CMD-DSG* "I"
company_address =	** address
company_email =	** email_address
company_fax_no =	** phone_no
company_name =	** {legal_character}
company_phone_no =	** phone_no
company_poc =	** name
completed_pr =	*purchase request that has been approved by the supervisor* @doc_ref_no + funded_pr + request_date + supervisorary_approval
comt_ref_no =	*SOMARDS document reference number* doc_ref_no
corrected_pr =	*purchase request that has been corrected by the requester* rejected_pr + corrections
corrections =	*corrections to a rejected purchase request* **TBD**
country =	** {alphabetic_character}

25

cum_btch_value =	*SOMARDS cumulative batch total for the days certification* *units: dollars*
date_required =	*date shipment is required by* date
delivery_date =	*estimated date for delivery from vendor* date
description =	*item description* {legal_character}
doc_ref_no =	*purchase request document reference number* "W-" + *TBD*
email_address =	** *TBD*
EMPLOYEE =	{employee}
employee =	*employee information - the bare minimum should contain* user)info + {roles}
eor =	*funding element of resource* {alphanumeric}
estimated_cost =	*estimated cost of an item* *units: dollars*
explanation =	*rejection, cancelation, or return explanation* {legal_character}
finished_pr =	*purchase request where the total actual cost is less than or equal to the total estimated cost* approved_pr + {actual_cost} + {item_tot_act_cost} + total_actual_cost
first_name =	*a person's first name* {alphabetic_character}
fund_source =	** jo_no + eor

fund_source_approval =   *budget analyst approval of fund source*
                                   ["Yes" | "No"]

funded_pr =   *purchase request with a fund source*
                                   partial_pr + fund_source

funds =   *funding information for SOMARDS certification*
                                   trns_cd + user_auth_key + cmd_dsg + update_code +
                                   blk_no + blk_tkt_dt + batch_no + rej_rept_director +
                                   doc_ref_no + jo_no + eor + act_amt

inbox =   *purchase requests requiring action from user*
                                   {active_pr}

inbox_inquiry =   **
                                   TBD

inbox_location =   *current purchase request location*
                                   TBD

item =   **
                                   @line_item_no + description + unit_of_issue + qty +
                                   estimated_cost + actual_cost + item_tag + acceptance +
                                   item_tot_est_cost + item_tot_act_cost

items =   **
                                   {item} + specifications

item_tag =   *property book officer inputs 'yes' item is taggable,
                                   receiving overwrites with bar_code_no*
                                   ["Yes" | bar_code_no]

item_tags =   {item_tags}

item_tot_act_cost =   *line item total actual cost*
                                   units: dollars

item_tot_est_cost =   *line item total estimated cost*
                                   units: dollars

jo_no =   *funding job number*
                                   {alphanumeric}

last_name =   *a person's last name*
                                   {alphabetic_character}

line_item_no =   *line item number*
                                  {numeric_digit}

mail_stop =   *mail stop or department*
                                  {legal_character}

name =   **
                                  first_name + last_name

nomenclature =   *SOMARDS certification comment field*
                                  {alphanumeric}

office_symbol =   *ARL office symbol*
                                  "AMSRL-" + {alphabetic_character} + "-" +
                                  {alphabetic_character}

orderable_pr =   *purchase request that can be ordered by procurement*
                                  taggable_pr + property_approval

ordered_pr =   *purchase request that has been ordered*
                                  finished_pr + delivery_date + vendor + po_number

partial_pr =   *purchase request with items and vendors filled in*
                                  blank_pr + date_required + priority_code + items + vendors

phone_no =   *a phone number*
                                  {numeric_digit}

po_number =   *purchase order number*
                                  TBD

postal_code =   *postal/zip code*
                                  {numeric_digit}

pr_details =   *details about the purchase request*
                                  TBD

priority_code =   *request priority code*
                                  range:01 - 15, 99
                                  {numeric_digit}

property_approval =   *property book officer approval*
                                  ["Yes" | "No"]

qty =	*quantity requested* {numeric_digit}
receipt =	*shipment receipt* ["Yes" | "No"]
received_pr =	*purchase request that has been received* ordered_pr + receipt
reconcile =	*end of the day SOMARDS reconcile info* trns_cd + user_auth_key + cmd_dsg + update_code + blk_no + blk_tkt_dt + batch_no + tot_blk + tot_batch + ty_act_cd + cum_btch_value + variance
rejected_pr =	*purchase request that has been rejected* active_pr + explanation
rej_rept_director =	*SOMARDS REJ-REPT-DIRECTOR* "R"
report =	** *TBD*
report_type =	*type of report to generate* *TBD*
request_date =	*date purchase request was approved by supervisor* date
requester_userid =	** userid
return_message =	*message returned from SOMARDS process* ["processing complete" | "bad user_auth_key" | "wrong update code" | "blk_no/blk_tkt_dt already exists" | "accounting class displayed" | "blk_no/blk_tkt_dt doesn't exist" | "invalid jo_no" | "invalid eor" | "insufficient funds" | "duplicate comt_ref_no" | "cum_btch_value" | "make changes" | "variance"]
role =	*user role* {alphanumeric}
room_no =	*room number* {alphanumeric}

sole_source_just =   *justification for using a single vendor*
                            {legal_character}

special_approval =   *approval from a special approving officials*
                            ["Yes" | "No"]

special_pr =   *purchase request with special approvals*
                            certified_pr + special_approval

state =   *state or province*
                            {legal_character}

status =   **
                            TBD

status_inquiry =   **
                            TBD

street_address =   **
                            {legal_character}

supervisory_approval =   *approval from supervisor*
                            ["Yes" | "No"]

taggable_pr =   *purchase request that has had item tags attached by
                            property book officer*
                            special_pr + item_tags

tagged_pr =   *purchase request that has been received and taggable items
                            have been appropriately tagged*
                            received_pr + item_tags

total_actual_cost =   *the total actual cost of the purchase request*
                            units: dollars

total_estimated_cost =   *the total estimated cost of the purchase request*
                            units: dollars

tot_batch =   *SOMARDS batch number*
                            units: dollars
                            ["0.00" | cum_btch_value]

tot_blk =   *SOMARDS total block*
                            units: dollars
                            ["0.00" | cum_btch_value]

trns_cd =   *SOMARDS transaction code*
                             ["003" | "004" | "310"]

ty_act_cd =   *SOMARDS action code*
                             "C"

unit_of_issue =   **
                             TBD

update_code =   *SOMARDS update code*
                             ["CM" | "NM"]

user_auth_key =   *SOMARDS user authorization key*
                             {alphanumeric}

user_info =   *user information*
                             name + office_symbol + phone_no + bldg_no + room_no

userid =   **
                             {alphanumeric}

variance =   *SOMARDS variance between tot_batch and
                             cum_btch_value - should be 0.00*
                             units: dollars

vendor =   *vendor information*
                             company_name + company_address + company_phone_no
                             + company_fax_no + company_poc + company_email

vendors =   *up to three suggested vendors*
                             {vendor} + sole_source_justification

INTENTIONALLY LEFT BLANK.

7. References

1. Business Process Reengineering Office. "To-Be Model: Formal Contracts." U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, October 1995.

2. McGurin, D. "Contract Req Software Development Plan." Enterprise Systems Division, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, to be published.

3. Yourdon, E. *Modern Structured Analysis*. Englewood Cliffs, NJ: Yourdon Press, 1989.

INTENTIONALLY LEFT BLANK.

NO. OF
COPIES ORGANIZATION

2 DEFENSE TECHNICAL
   INFORMATION CENTER
   DTIC DDA
   8725 JOHN J KINGMAN RD
   STE 0944
   FT BELVOIR VA 22060-6218

1 HQDA
   DAMO FDQ
   D SCHMIDT
   400 ARMY PENTAGON
   WASHINGTON DC 20310-0460

1 OSD
   OUSD(A&T)/ODDDR&E(R)
   R J TREW
   THE PENTAGON
   WASHINGTON DC 20301-7100

1 DPTY CG FOR RDE HQ
   US ARMY MATERIEL CMD
   AMCRD
   MG CALDWELL
   5001 EISENHOWER AVE
   ALEXANDRIA VA 22333-0001

1 INST FOR ADVNCD TCHNLGY
   THE UNIV OF TEXAS AT AUSTIN
   PO BOX 202797
   AUSTIN TX 78720-2797

1 DARPA
   B KASPAR
   3701 N FAIRFAX DR
   ARLINGTON VA 22203-1714

1 NAVAL SURFACE WARFARE CTR
   CODE B07 J PENNELLA
   17320 DAHLGREN RD
   BLDG 1470 RM 1101
   DAHLGREN VA 22448-5100

1 US MILITARY ACADEMY
   MATH SCI CTR OF EXCELLENCE
   DEPT OF MATHEMATICAL SCI
   MAJ M D PHILLIPS
   THAYER HALL
   WEST POINT NY 10996-1786

NO. OF
COPIES ORGANIZATION

1 DIRECTOR
   US ARMY RESEARCH LAB
   AMSRL DD
   J J ROCCHIO
   2800 POWDER MILL RD
   ADELPHI MD 20783-1145

1 DIRECTOR
   US ARMY RESEARCH LAB
   AMSRL CS AS (RECORDS MGMT)
   2800 POWDER MILL RD
   ADELPHI MD 20783-1145

3 DIRECTOR
   US ARMY RESEARCH LAB
   AMSRL CI LL
   2800 POWDER MILL RD
   ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

4 DIR USARL
   AMSRL CI LP (305)

NO. OF
COPIES ORGANIZATION

8   DIRECTOR
      US ARMY RESEARCH LAB
      AMSRL IS
      J GRILLS
      AMSRL CI A
      R ROSEN (2 CPS)
      AMSRL CS
      COL K LOGAN
      AMSRL CI E
      W JERNIGAN
      W MCCOY
      B SCHALLHORN
      AMSRL SE S
      A SINDORIS
      2800 POWDER MILL RD
      ADELPHI MD 20783-1145

1   DIRECTOR
      US ARMY RESEARCH LAB
      AMSRL EA
      P STAY
      WSMR NM 88002-5501

      ABERDEEN PROVING GROUND

14   DIR USARL
      AMSRL CI
        N RADHAKRISHNAN
      AMSRL CI E
        D ULERY (10 CPS)
      AMSRL HR SF
        B AMREIN
      AMSRL WM IM
        R MCGEE
      AMSRL CS AP
        D ORE

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project(0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1999	September 1997 - January 1998

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Contract Req Software Requirements Analysis Version 1: A C-BASS Component	AC9UA4EC

6. AUTHOR(S)

Denis McGurin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Research Laboratory ATTN: AMSRL-CI-E Aberdeen Proving Ground, MD 21005-5067	ARL-MR-458

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)	10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited.	

13. ABSTRACT (Maximum 200 words)

This document contains the software requirements analysis for a prototype of Contract Request Version 1.0 (Contract Req). As a component of the Corporate Business Application Software System (C-BASS), this application automates the preparation of the requester's Procurement Data Package (PDP). The document follows the process of structured analysis, or step-wise refinement of requirements, as applied to the development of Contract Req. The "environmental model" includes a high-level system description, followed by a context diagram and a list of events to which the system must respond. The "behavioral model" includes a data flow diagram (DFD) for each of the five Contract Req subsystems. From this representation, the basic functional specifications are derived and represented in structured English (or program design language). The final segment of the document includes a data dictionary.

14. SUBJECT TERMS		15. NUMBER OF PAGES
Contract preparation, formal contracts, user requirements, structured analysis, data flow diagrams, data dictionary		39
		16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18   298-102

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author_____ARL-MR-458 (McGurin)_____ Date of Report _August 1999_____

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

_____

_____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

_____

_____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

_____

_____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

_____

_____

_____

CURRENT
ADDRESS

Organization
_____

Name   E-mail Name
_____

Street or P.O. Box No.
_____

City, State, Zip Code
_____

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD
ADDRESS

Organization
_____

Name
_____

Street or P.O. Box No.
_____

City, State, Zip Code
_____

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

BUSINESS REPLY MAIL FIRST CLASS PERMIT NO 0001,APG,MD	**NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES**

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
US ARMY RESEARCH LABORATORY
ATTN AMSRL CI E
ABERDEEN PROVING GROUND MD 21005-5067